

Hi,

as in railML 2.x we also need to specify times in railML 3.x. As you probably all know we are usually not just dealing with a single arrival or departure time, but usually there is operational times as well as commercial times.

In railML 2.x we've been describing the kind of time provided in a `//ocpTT/times` using the attribute `scope` and its enum values. It allowed for the values: `actual`, `calculated`, `published`, `scheduled`, `earliest` and `latest`. Reviewing the usage of these values we found that some of them were rather unclear regarding when to use them, or that they actually were not used at all as per our knowledge. In that light we don't want to simply take over the values we have had in railML 2.x but rather define the ones which are really needed.

Additionally we decided in railML 3.x to separate the description of recorded data from planned data, means a specific train on a specific day (actual train data) that is to be exchanged via railML in order to communicate estimates or actual recorded arrival and departure data is modelled differently from trains that run like once a day (compressed schedule data). Thus the enum value `'actual'` would not make sense in an enum that is used in railML 3.x only to describe compressed schedule data.

For railML 3.2 we so far are focussing on compressed schedule data only. We are developing the model along the requirements of the use case ITMS, so a traffic management system. The timetable developer group so far found that for this we would certainly need the time scope values `'commercial'` and `'operational'`. `'commercial'` being that time communicated to the passenger by printed schedules, online schedule, a.s.o, while `'operational'` is the actual operational times used for internal scheduling purposes.

This post is to ask you, what other kind of times you would consider required for compressed schedule data of a TMS. Please let us know, ideally with a short description of what to express with that, so we can consider that for the development of railML 3.2.

Thanks for your support.

wie in railML 2.x müssen wir auch in railML 3.x Zeiten angeben. Wie Sie sicher alle wissen, haben wir es in der Regel nicht nur mit einer einzelnen Ankunfts- oder Abfahrtszeit zu tun, sondern in der Regel gibt es sowohl betriebliche als auch kommerzielle Zeiten.

In railML 2.x haben wir die Art der Zeit, die in einem `//ocpTT/times` bereitgestellt wird, mit dem Attribut `scope` und seinen Enum-Werten beschrieben. Es erlaubte die Werte: `actual`, `calculated`, `published`, `scheduled`, `earliest` und `latest`. Bei der Überprüfung der Verwendung dieser Werte stellten wir fest, dass die Verwendung einiger dieser Werte ziemlich unklar war, oder dass sie unseres Wissens nach gar nicht verwendet wurden. Vor diesem Hintergrund wollen wir nicht

einfach die Werte aus railML 2.x übernehmen, sondern die Werte definieren, die wirklich benötigt werden.

Zusätzlich haben wir uns in railML 3.x entschieden, die Beschreibung von aufgezeichneten Daten von geplanten Daten zu trennen, d.h. ein bestimmter Zug an einem bestimmten Tag (Ist-Zugdaten), der über railML ausgetauscht werden soll, um Prognosen oder tatsächlich aufgezeichnete Ankunfts- und Abfahrtszeiten zu kommunizieren, wird anders modelliert als Züge, die etwa einmal am Tag fahren (komprimierte Fahrplandaten). Daher würde der Enum-Wert 'actual' in einem Enum, das in railML 3.x nur zur Beschreibung komprimierter Fahrplandaten verwendet wird, keinen Sinn machen.

Für railML 3.2 fokussieren wir uns zunächst nur auf komprimierte Fahrplandaten. Wir entwickeln das Modell nach den Anforderungen des Use Cases ITMS, also eines Traffic Management Systems. Die Fahrplanentwickler-Gruppe hat bisher festgestellt, dass wir dafür auf jeden Fall die Zeitklassen 'commercial' und 'operational' benötigen. 'commercial' sind die Zeiten, die dem Fahrgast durch gedruckte Fahrpläne, Online-Fahrplan usw. mitgeteilt werden, während 'operational' die tatsächlichen Betriebszeiten sind, die für interne Planungszwecke verwendet werden.

Mit diesem Post möchten wir Sie bitten, uns mitzuteilen, welche andere Art von Zeiten Sie für die komprimierten Fahrplandaten eines TMS als erforderlich ansehen würde, idealerweise mit einer kurzen Beschreibung, was damit ausgedrückt werden soll, damit wir das bei der Entwicklung von railML 3.2 berücksichtigen können.

Vielen Dank für Ihre Unterstützung.

Best regards, Milan
